MIIA0106

Python and C Programming Language

อาจารย์ สุทิศ องอาจ

ภาคการศึกษา 2 ปีการศึกษา 2567

MIIA0106	Python and C Programming Language		SUN-เช้า-23/03/2568
A	MON/2	D604	[CPA/1,CPAI/1,EMAT/1]#ALL#
	MON/3	MII203	
	MON/3	MII202 A	
	MON/3	MII202 B	
B	SAT/2	MII207 B	[CPA+/1,EMAT+/1,EMA/2]#ALL#
	SUN/1	MII202 A	
	SUN/1	MII202 B	
EP	- /0	-	

Update V1 2024-11-16

Update V2 2024-11-20

Update V3 2024-11-30

Update V4 2024-12-07

Update V5 2024-12-14

5. คาบที่ 6: Array, Pointer (6 ชม.)

5.1. บรรยาย (3 ชม.)

5.1.1.ทำไหมต้องใช้ อาร์เรย์ (Array)

เขียนโปรแกรมเพื่อจัดการคะแนนของนักเรียน 5 คน โดยมีรายละเอียดดังนี้:

1) รับคะแนนจากผู้ใช้
2) คำนวณผลรวมของคะแนน
3) คำนวณค่าเฉลี่ยของคะแนน
4) แสดงผลคะแนนของนักเรียนแต่ละคน, ผลรวมคะแนน, และค่าเฉลี่ย

วิธีทำ

1) วัตถุประสงค์ของการเขียนโปรแกรม

   เพื่อสร้างโปรแกรมที่รับคะแนนนักเรียน 5 คนจากผู้ใช้ และคำนวณผลรวมและค่าเฉลี่ยของคะแนน พร้อมทั้งแสดงผลคะแนนของนักเรียนแต่ละคน ผลรวม และค่าเฉลี่ย

2) รูปแบบผลลัพธ์ที่ต้องการ

```
Enter score of Student 1: 85
Enter score of Student 2: 90
Enter score of Student 3: 78
Enter score of Student 4: 92
Enter score of Student 5: 88
Scores of Students:
Student 1: 85
Student 2: 90
Student 3: 78
Student 4: 92
Student 5: 88
Total Score: 433
Average Score: 86.6
```

3) ข้อมูลนำเข้า

คะแนนของนักเรียน 5 คน (รับจากผู้ใช้)

4) ตัวแปรที่ใช้

C++:

- scores[5]: Array เก็บคะแนนของนักเรียน
- totalScore: ตัวแปรเก็บผลรวมคะแนน
- averageScore: ตัวแปรเก็บค่าเฉลี่ย

Python:

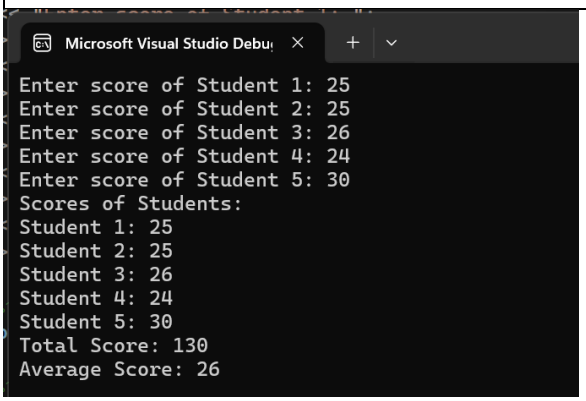
- scores: List เก็บคะแนนของนักเรียน
- total_score: ตัวแปรเก็บผลรวมคะแนน
- average_score: ตัวแปรเก็บค่าเฉลี่ย

5) วิธีการประมวลผล

1) รับค่าคะแนนนักเรียน 5 คนจากผู้ใช้
2) เก็บค่าคะแนนใน Array (C++) หรือ List (Python)
3) คำนวณผลรวมคะแนนโดยใช้ลูป
4) คำนวณค่าเฉลี่ยโดยหารผลรวมด้วยจำนวนคะแนนทั้งหมด
5) แสดงผลคะแนนแต่ละคน, ผลรวม, และค่าเฉลี่ย

ก่อนใช้ Array (C++)

```cpp
#include <iostream>
using namespace std;

int main() {
    // เก็บคะแนนนักเรียน 5 คน
    int score1, score2, score3, score4, score5;
    cout << "Enter score of Student 1: ";
    cin >> score1;
    cout << "Enter score of Student 2: ";
    cin >> score2;
    cout << "Enter score of Student 3: ";
    cin >> score3;
    cout << "Enter score of Student 4: ";
    cin >> score4;
    cout << "Enter score of Student 5: ";
    cin >> score5;

    // คำนวณผลรวมคะแนน
    int totalScore = score1 + score2 + score3 + score4 + score5;

    // คำนวณค่าเฉลี่ย
    double averageScore = totalScore / 5.0;

    // แสดงผลลัพธ์
    cout << "Scores of Students:" << endl;
    cout << "Student 1: " << score1 << endl;
    cout << "Student 2: " << score2 << endl;
    cout << "Student 3: " << score3 << endl;
    cout << "Student 4: " << score4 << endl;
    cout << "Student 5: " << score5 << endl;
    cout << "Total Score: " << totalScore << endl;
    cout << "Average Score: " << averageScore << endl;

    return 0;
}
```



ก่อนใช้ Array (Python)

```python
def main():
    # รับคะแนนนักเรียน 5 คน
    score1 = int(input("Enter score of Student 1: "))
    score2 = int(input("Enter score of Student 2: "))
    score3 = int(input("Enter score of Student 3: "))
    score4 = int(input("Enter score of Student 4: "))
    score5 = int(input("Enter score of Student 5: "))
    # คำนวณผลรวมคะแนน
    total_score = score1 + score2 + score3 + score4 + score5
    # คำนวณค่าเฉลี่ย
    average_score = total_score / 5
    # แสดงผลลัพธ์
    print("Scores of Students:")
    print("Student 1:", score1)
    print("Student 2:", score2)
    print("Student 3:", score3)
    print("Student 4:", score4)
    print("Student 5:", score5)
    print("Total Score:", total_score)
    print("Average Score:", average_score)

if __name__ == "__main__":
    main()
```

```
PS C:\Users\Maori\Desktop\TEST_lab\python> & C:/Use
p/TEST_lab/python/test.py
Enter score of Student 1: 35
Enter score of Student 2: 25
Enter score of Student 3: 12
Enter score of Student 4: 25
Enter score of Student 5: 35
Scores of Students:
Student 1: 35
Student 2: 25
Student 3: 12
Student 4: 25
Student 5: 35
Total Score: 132
Average Score: 26.4
PS C:\Users\Maori\Desktop\TEST_lab\python>
```

หลังใช้ Array (C++)

```cpp
#include <iostream>
using namespace std;

int main() {
	// Array เก็บคะแนนนักเรียน
	int scores[5];
	int totalScore = 0;
	double averageScore;

	// รับค่าคะแนนจากผู้ใช้
	for (int i = 0; i < 5; i++) {
		cout << "Enter score of Student " << (i + 1) << ": ";
		cin >> scores[i];
		totalScore += scores[i]; // คำนวณผลรวมคะแนน
	}

	// คำนวณค่าเฉลี่ย
	averageScore = totalScore / 5.0;

	// แสดงผลลัพธ์
	cout << "Scores of Students:" << endl;
	for (int i = 0; i < 5; i++) {
		cout << "Student " << (i + 1) << ": " << scores[i] << endl;
	}
	cout << "Total Score: " << totalScore << endl;
	cout << "Average Score: " << averageScore << endl;

	return 0;
}
```

หลังใช้ Array (List)

```python
def main():
    # List เก็บคะแนนนักเรียน
    scores = []
    total_score = 0

    # รับค่าคะแนนจากผู้ใช้
    for i in range(5):
        score = int(input(f"Enter score of Student {i + 1}: "))
        scores.append(score)
        total_score += score  # คำนวณผลรวมคะแนน

    # คำนวณค่าเฉลี่ย
    average_score = total_score / len(scores)

    # แสดงผลลัพธ์
    print("Scores of Students:")
    for i, score in enumerate(scores):
        print(f"Student {i + 1}: {score}")
    print("Total Score:", total_score)
    print("Average Score:", average_score)

if __name__ == "__main__":
    main()
```

สรุป

- **โค้ดก่อนใช้ Array:**
    - เหมาะสำหรับข้อมูลจำนวนน้อย เช่น 5 คน
    - หากเพิ่มจำนวนนักเรียน จะทำให้ต้องสร้างตัวแปรเพิ่มและแก้ไขโค้ดมาก
- **โค้ดหลังใช้ Array:**
    - รองรับการจัดการข้อมูลจำนวนนักเรียนมากขึ้น
    - โค้ดกระชับและยืดหยุ่นมากกว่า โดยใช้ลูปจัดการข้อมูล

การใช้ Array จึงช่วยเพิ่มประสิทธิภาพและลดข้อผิดพลาดที่อาจเกิดจากการสร้างตัวแปรซ้ำซ้อน

5.1.2.Array (อาร์เรย์) คืออะไร การประกาศและการใช้งาน

Array (อาร์เรย์) คืออะไร

1) เป็นโครงสร้างข้อมูลประเภทหนึ่งที่ใช้เก็บ ชุดของข้อมูลหลายค่า ภายในตัวแปรเดียว
2) ข้อมูลที่เก็บในอาร์เรย์จะต้องเป็นชนิดข้อมูลเดียวกัน เช่น ตัวเลข หรือข้อความ
3) สามารถเข้าถึงค่าที่อยู่ในอาร์เรย์แต่ละตำแหน่งได้โดยใช้ดัชนี (Index) ซึ่งเริ่มต้นที่ 0

การประกาศและการใช้งาน Array

รูปแบบ:

```
data_type array_name[array_size];

data_type คือชนิดข้อมูล เช่น int, float, char
array_name คือชื่อของอาร์เรย์
array_size คือจำนวนของข้อมูลที่อาร์เรย์สามารถเก็บได้
```

ตัวอย่างการประกาศและใช้งาน

```cpp
#include <iostream>
using namespace std;

int main() {
    // ประกาศอาร์เรย์ขนาด 5 สำหรับเก็บตัวเลขจำนวนเต็ม
    int numbers[5] = { 10, 20, 30, 40, 50 };

    // เข้าถึงและแสดงค่าจากอาร์เรย์
    cout << "First element: " << numbers[0] << endl; // ดัชนีเริ่มที่ 0
    cout << "Second element: " << numbers[1] << endl;

    // ใช้ลูปแสดงค่าทั้งหมดในอาร์เรย์
    cout << "All elements: ";
    for (int i = 0; i < 5; i++) {
        cout << numbers[i] << " ";
    }
    cout << endl;

    return 0;
}
```

ใน **Python**

การใช้งาน **List (คล้าย Array ใน Python)**

ใน Python ไม่มี Array แบบใน C++ แต่จะใช้ **List** ซึ่งมีความยืดหยุ่นและสามารถเก็บค่าหลายประเภทได้

ตัวอย่างการประกาศและใช้งาน

```
# ประกาศ List และกำหนดค่าเริ่มต้น
numbers = [10, 20, 30, 40, 50]

# เข้าถึงและแสดงค่าจาก List
print("First element:", numbers[0])  # ดัชนีเริ่มที่ 0
print("Second element:", numbers[1])

# ใช้ลูปแสดงค่าทั้งหมดใน List
print("All elements:", end=" ")
for number in numbers:
    print(number, end=" ")
print()
```

การเปรียบเทียบการใช้งาน

คุณสมบัติ	Array (C++)	List (Python)
ชนิดข้อมูล	ต้องเป็นชนิดเดียวกัน	เก็บข้อมูลได้หลายชนิดในตัวเดียว
การกำหนดขนาด	ต้องกำหนดขนาดตอนประกาศ	ไม่ต้องกำหนดขนาด (ยืดหยุ่น)
การเข้าถึงข้อมูล	ใช้ดัชนี (array[index])	ใช้ดัชนี (list[index])
การประมวลผล	ใช้ลูป for หรือ while	ใช้ลูป for หรือฟังก์ชันสำเร็จรูป เช่น sum

5.1.3.ประเภทของ Array

Array สามารถแบ่งออกได้หลายประเภทตามลักษณะการใช้งานและโครงสร้างข้อมูล โดยทั่วไปสามารถจำแนกได้ดังนี้:

1. แบ่งตามมิติของ Array

1.1. Array 1 มิติ (One-Dimensional Array)

- เป็น Array ที่มีข้อมูลอยู่ในลำดับเดียว (เหมือนเส้นตรง)
- ตัวอย่าง: เก็บคะแนนของนักเรียน 5 คนในรูปแบบ 1 แถว

```
C++
   int scores[5] = { 85, 90, 78, 92, 88 };
Python
scores = [85, 90, 78, 92, 88]
```

1.2. Array 2 มิติ (Two-Dimensional Array)

เป็น Array ที่มีข้อมูลในรูปแบบแถวและคอลัมน์ (เหมือนตาราง)

ตัวอย่าง: เก็บคะแนนของนักเรียน 3 คนใน 4 วิชา

```
C++
   int scores[3][4] = {
   {85, 90, 78, 92},  // คะแนนนักเรียนคนที่ 1
   {88, 76, 93, 85},  // คะแนนนักเรียนคนที่ 2
   {91, 82, 88, 95}   // คะแนนนักเรียนคนที่ 3
   };
Python
scores = [
    [85, 90, 78, 92],  # คะแนนนักเรียนคนที่ 1
    [88, 76, 93, 85],  # คะแนนนักเรียนคนที่ 2
    [91, 82, 88, 95]   # คะแนนนักเรียนคนที่ 3
]
```

1.3. Array หลายมิติ (Multi-Dimensional Array)

เป็น Array ที่มีมิติมากกว่า 2 เช่น 3 มิติ, 4 มิติ

ใช้สำหรับจัดเก็บข้อมูลที่ซับซ้อน เช่น พิกัดในพื้นที่ 3 มิติ

```
C++
    int arr[2][3][4]; // Array 3 มิติ ขนาด 2x3x4


Python
arr = [
    [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]],
    [[13, 14, 15, 16], [17, 18, 19, 20], [21, 22, 23, 24]]
]
```

2. แบ่งตามลักษณะการจัดการหน่วยความจำ

2.1. Static Array

ขนาดของ Array ถูกกำหนดตั้งแต่ตอนประกาศและไม่สามารถเปลี่ยนแปลงได้ในระหว่างการทำงานของโปรแกรม

ใช้ในกรณีที่ทราบขนาดของข้อมูลล่วงหน้า

```
    int arr[2][3][4]; // Array 3 มิติ ขนาด 2x3x4
```

2.2. Dynamic Array

ขนาดของ Array สามารถเปลี่ยนแปลงได้ในระหว่างการทำงานของโปรแกรม

เหมาะสำหรับการจัดการข้อมูลที่ขนาดไม่แน่นอน

```
#include <iostream>
using namespace std;

int main() {
    int size;
    cout << "Enter size of array: ";
    cin >> size;

    int* arr = new int[size]; // สร้าง Dynamic Array
```

```
    for (int i = 0; i < size; i++) {
        arr[i] = i + 1; // เพิ่มค่าใน Array
    }

    // แสดงค่าจาก Array
    for (int i = 0; i < size; i++) {
        cout << arr[i] << " ";
    }
    delete[] arr; // ลบหน่วยความจำ
    return 0;
}
```

คำอธิบายคำสั่ง

1. **int***
    - เป็นตัวชี้ (Pointer) ที่ชี้ไปยังตำแหน่งหน่วยความจำที่เก็บข้อมูลประเภท int
2. **new int[size]**
    - เป็นคำสั่งสำหรับการจองหน่วยความจำ (Memory Allocation) แบบ Dynamic ใน Heap Memory
    - ขนาดของ Array ถูกกำหนดด้วย size
    - จะคืนค่าที่อยู่หน่วยความจำ (Pointer) ของ Array ที่สร้างขึ้น
3. **arr**
    - ตัวแปร Pointer (int*) ที่เก็บตำแหน่งหน่วยความจำที่ new int[size] คืนค่าให้

ตัวแปร arr จึงสามารถใช้งานเหมือน Array ปกติผ่านการเข้าถึงด้วยดัชนี (Index) เช่น arr[0], arr[1] ฯลฯ

```
def main():
   # List ใน Python เป็น Dynamic Array
    arr = []
    size = int(input("Enter size of array: "))

    for i in range(size):
        arr.append(i + 1)  # เพิ่มค่าใน List

    print("Array:", arr)


if __name__ == "__main__":
    main()
```

เปรียบเทียบวิธีการ

วิธี	ข้อดี	ข้อเสีย
Static Array	ประสิทธิภาพสูง, ใช้งานง่าย	ขนาดคงที่ ไม่สามารถเปลี่ยนแปลงได้
Dynamic Array (int*)	ขนาดเปลี่ยนได้ตามต้องการ, ประหยัดหน่วยความจำ	ต้องจัดการหน่วยความจำเอง (new, delete)
std::vector	ใช้งานง่าย, ปลอดภัย, ยืดหยุ่นสูง	อาจใช้หน่วยความจำมากกว่า Static Array เล็กน้อย

5.2. ปฏิบัติ (3 ชม.)

1.เขียนโปรแกรมเพื่อจัดการคะแนนของนักเรียน **10** คน โดยมีรายละเอียดดังนี้: (ห้ามใช้ **Array**)

1) รับคะแนนจากผู้ใช้

2) คำนวณผลรวมของคะแนน

3) คำนวณค่าเฉลี่ยของคะแนน

4) แสดงผลคะแนนของนักเรียนแต่ละคน, ผลรวมคะแนน, และค่าเฉลี่ย

1.วัตถุประสงค์ของการเขียนโปรแกรม

2.รูปแบบผลลัพธ์ที่ต้องการ

3.ข้อมูลนำเข้า

4.ตัวแปรที่ใช้

5.วิธีการประมวลผล

6 code

C++

Python

7.ผลการรัน

2.เขียนโปรแกรมเพื่อจัดการคะแนนของนักเรียน 10 คน โดยมีรายละเอียดดังนี้: (โดยใช้ Array)

1) รับคะแนนจากผู้ใช้

2) คำนวณผลรวมของคะแนน

3) คำนวณค่าเฉลี่ยของคะแนน

4) แสดงผลคะแนนของนักเรียนแต่ละคน, ผลรวมคะแนน, และค่าเฉลี่ย

1.วัตถุประสงค์ของการเขียนโปรแกรม

2.รูปแบบผลลัพธ์ที่ต้องการ

3.ข้อมูลนำเข้า

4.ตัวแปรที่ใช้

5.วิธีการประมวลผล

6 code

C++

Python

7.ผลการรัน

3.เขียนโปรแกรมเพื่อรับจำนวน **20 จำนวน** โดยมีรายละเอียดดังนี้: (โดยใช้ Array)

1) รับจำนวนทศนิยม จากผู้ใช้

2) คำนวณรวม ของผลลบ (ตัวที่ 1 –ตัวที่ 2 ... -ตัวที่ 20)

3) คำนวณผลรวม ของคูณ

4) แสดงผลจำนวน และผลการลบ ผลการคูณ

1.วัตถุประสงค์ของการเขียนโปรแกรม

2.รูปแบบผลลัพธ์ที่ต้องการ

3.ข้อมูลนำเข้า

4.ตัวแปรที่ใช้

5.วิธีการประมวลผล

6 code

C++

Python

7.ผลการรัน